

Abstract. *The new economic and social dynamics increase project complexity and makes scheduling problems more difficult, therefore scheduling requires more versatile solutions as Multi Agent Systems (MAS). In this paper the authors analyze the implementation of a Multi-Agent System (MAS) considering two scheduling problems: TCPSP (Time-Constrained Project Scheduling), and RCPSP (Resource-Constrained Project Scheduling). The authors propose an improved BDI (Beliefs, Desires, and Intentions) model and present the first the MAS implementation results in JADE platform*

Keywords: multi-agent architecture, scheduling, project management, BDI architecture, JADE.

A MULTI-AGENT SYSTEM WITH APPLICATION IN PROJECT SCHEDULING

Constanța Nicoleta BODEA

*The Bucharest Academy of Economic Studies
6, Romana Square, 1st district, Bucharest,
Romania*

Email: bodea@ase.ro

Ileana Ruxandra BADEA

*The Bucharest Academy of Economic Studies
6, Romana Square, 1st district, Bucharest,
Romania*

Email: badea_ruxandra@yahoo.com

Radu-Ioan MOGOȘ

*The Bucharest Academy of Economic Studies
6, Romana Square, 1st district, Bucharest,
Romania*

Email: mogos.radu@gmail.com

Management & Marketing

*Challenges for the Knowledge Society
(2011) Vol. 6, No. 4, pp. 573-590*

1. The project scheduling problem

The traditional place of the scheduling is between planning and execution. It produces decisions on the specific resources, operations, and their timing to perform the activities (the schedule). Project scheduling, as a NP-complete problem, is a difficult task for the human planners and schedulers (Hurink, Kok, Paulus and Schutten, 2009). A project scheduling problem is normally characterized by objective functions, features of resources, and pre-emptive conditions (Lee and Kim, 1996). Minimizing of project duration is often used as the objective function, while other objectives such as minimization of total project cost and levelling of resource usage are also considered. For many projects there is a trade-off between project cost and project duration.

We propose an agent-based approach for solving the *Time-Constrained Project Scheduling Problem (TCPSP)* extended with cost optimization criteria, which is a specific scheduling problem. The TCPSP derives from the well studied *RCPSP (Resource-Constrained Project Scheduling Problem)* (Kolisch and Padman, 2001). However, there are fundamental differences between the time-constrained and the resource-constrained variant. In the first, the deadlines cannot be exceeded and resource profiles may be changed, whereas in the second, the resource availability cannot be exceeded. Moreover, in the TCPSP, a non-regular objective function is considered. Therefore, most existing solution techniques of the RCPSP are not suitable for the TCPSP. Different approaches to project planning and scheduling have been developed (Bodea and Niculescu, 2006).

In our model, as it was presented also in (Bodea, Badea and Purnus, 2010), a single project consists of a set $A = \{0, 1, n+1\}$ of activities which have to be processed according to precedence constraints and overall delays for activities should not exceed proposed limits. Resources R_j for activity j are considered sufficient and the only constraint regarding resources is for activity buffers not to exceed an established limit of extra-costs. The objective of TCSP is to find the precedence completion times for all the activities such that the duration of the project to be minimized. A schedule, S is given by a vector of Finish times $\{F_1, F_2, \dots, F_n\}$. The ILP (Integer Linear Problem) format for TCSP is as following:

$$\text{Min } (F_{n+1}) \quad n+1 = \text{final activity of the project} \quad (1)$$

$$F_h \leq F_j - D_j, \quad j=1, \dots, n+1, \quad h=1, \dots, j-1,$$

$$D_j = \text{duration for activity } j \quad (2)$$

$$\sum_j [\text{Cost } (B_j)] \leq C, \quad j=1, \dots, n+1,$$

$$C = \text{cost limit for buffers regarding resources} \quad (3)$$

$$F_j \geq 0, D_j \geq 0, B_j \geq 0 \quad j=1, \dots, n+1 \quad (4)$$

In order to solve this problem, we consider the TCPSP with working in overtime, and hiring in regular time and in overtime. Relations (1) and (4) can be mixed in one function, in order to minimize a total cost of the project, including time and cost resources for activity buffers. This scheduling problem uses the time buffer notion and can be seen as a variant of the time-cost trade off scheduling problem.

Our model requires a dynamic and flexible solution to provide optimum results despite favorable or unfavorable input data. Classical scheduling solutions as: Critical Path Method, Program Evaluation and Review Technique often provide local optimum instead of the global one. The most appropriate solutions for multiple criteria scheduling problems are now considered Artificial Intelligence methods (AI) (Russell and Norvig, 2003).

2. The multi-agent system architecture

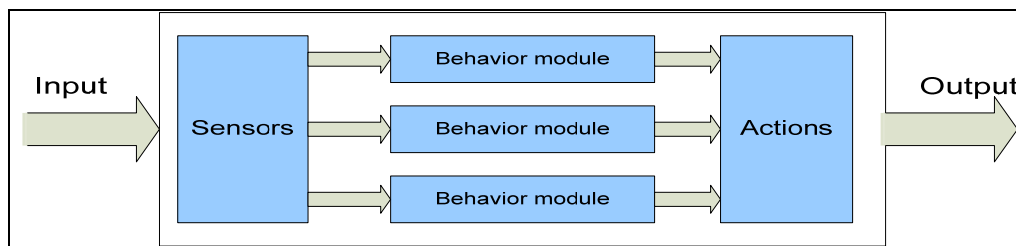
The modern approach to AI is centered on the concept of a rational agent. According to (Russell and Norvig, 2003), an agent is anything that can perceive its environment through sensors and act upon that environment through actuators. An agent that always tries to optimize an appropriate performance measure (expected utility) is called a rational agent. Such a definition of a rational agent is fairly general and can include human agents (having eyes as sensors, hands as actuators), robotic agents (having cameras as sensors, wheels as actuators), or software agents (having a graphical user interface as sensor and as actuator). However, agents are seldom stand-alone systems. In many situations they coexist and interact with other agents in several different ways. Such a system that consists of a group of agents that can potentially interact with each other is called a multi-agent system (MAS), and the corresponding subfield of AI that deals with principles and design of multi-agent systems is called distributed AI.

A MAS can be defined as a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver (Duffee, Lesser 1989). The agent research community has developed a number of models of interaction including coordination (Jennings, 1993; Duffee, 1999), collaboration (Cohen and Levesque, 1990, Pynadath and Tambe, 2002), and negotiation (Rosenschein and Zlotkin, 1994, Jennings et al., 2001, Kraus,

2001). The start point for creation of MAS is to define its architecture, then use of a methodology to design it and finally to implement it on a MAS platform.

Regarding the MAS architectures, there are agent level and system level architectures. At the agent level, there exist some classical models, such as: agent planning (STRIPS) architecture (Fikes and Nilsson, 1971), Beliefs, Desires, Intentions (BDI) architecture (Rao and Georgeff, 1995), reactive architecture (Brooks, 1986; Chapman and Agre, 1986; Maes, 1990; Kaelbling and Rosenschein, 1990), hybrid architectures, such as: Touring Machine (Ferguson, 1992) and Inter Rap (Müller Pischel, 1994).

Wooldridge & Jennings (1994) describe the **reactive architecture** as an architecture that does not have a central world model and does not use complex reasoning (Figure 1). Unlike knowledge-based agents that have an internal symbolic model from which to work, reactive agents act by stimulus-response to environmental states. The agent perceives an environmental change and reacts accordingly. Reactive agents can also react to messages from other agents. By adding a knowledge base to a simple reactive agent, you have an agent capable of making decisions that take into account previously encountered state information.



Source: Palensky, 1999.

Figure 1. The general architecture of a reactive agent

By adding goals and a planning mechanism, you can create a rather complex goal directed agent. One of the more elaborate uses of reactive agents was seen in Brook's Subsumption Architecture (Brooks, 1986) which is based on Brook's belief that the Artificial Intelligence community need not build "human level" intelligence directly into machines. The class diagram for a reactive agent is shown by figure 2.

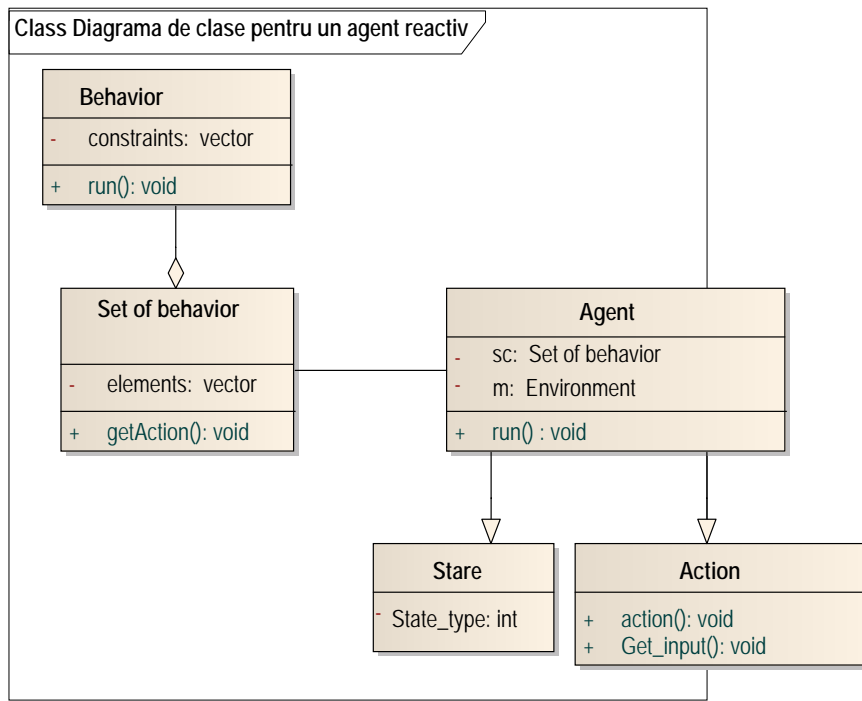
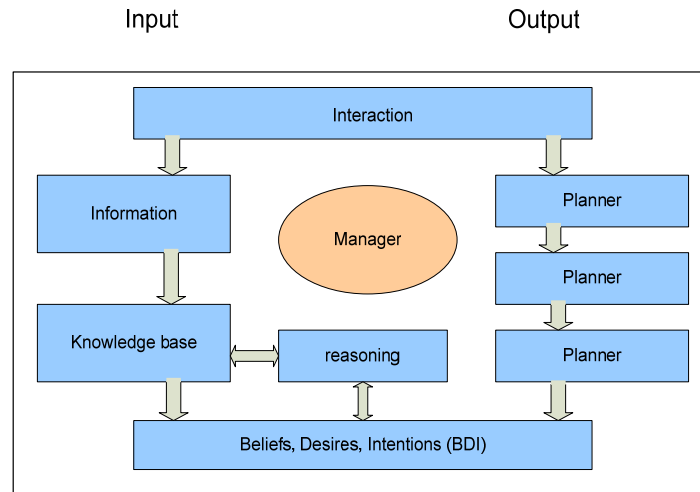


Figure 2. The class diagram for a reactive agent

The hybrid architecture obvious approach is to build an agent out of two or more subsystems:

- a deliberative one, containing a symbolic world model, which develops plans and makes decisions in the way proposed by symbolic AI;
- a reactive one, which is capable of reacting to events without complex reasoning.

Figure 3 presents the general architecture of a deliberative agent.



Source: Brenner, Zarnekow and Witting, 1998.

Figure 3. The general architecture of a deliberative agent

The MAS architectures based on direct communication include: contract based architectures and common specifications architectures. When systems are very complex, these types of architectures cannot produce good results, therefore, indirect communication through “federal system” is more appropriate.

We consider MAS to be a large-scale solution used for scheduling problems in several areas, such as:

- Space exploring missions. NASA is investigating the possibility of making space missions more autonomous, giving them richer decision making capabilities and responsibilities (Seah, Sierhuis and Clancey, 2011).
- Sociology. The MAS provide a novel tool for simulating societies (Davidsson, 2000).
- Experimental economics. The MAS can simulate macro and micro economic phenomena (Heckbert, 2009).
- E-medicine, e-diagnosis. The MAS can be used to simulate specific medical activities as diagnosis (Hussain and Wood, 2009).
- Internet. The MAS used as news readers, web browsers (google), mail readers, e-commerce, (Dawid, 1999), the first Double Auction market based on computerized agents (Chen, 2000).
- Simulated multi-agent traffic environment. The Autodrive architecture enables agents to plan routes in this environment (Wood, 1990).

The Autodrive architecture is centered on a planner which integrates traditional problem solving (the generation of hierarchical, temporally ordered route plans) with a process called *dynamic goal creation*: the continual run-time creation and modification of planning sub-goals which, while aiming to address the planner’s changing situational constraints, simultaneously strives to bring the agent nearer to its

goal. At this architecture we consider the importance of hierarchy in planning and scheduling, and implement it through specialized mediators. For each mentioned area project scheduling has specific implementation. We admit a general TCSP problem without concerning about project type and extend it with cost constraints if necessary.

3. The architecture of BDI

The BDI architecture (Figure 4) considers the human activity model to reach a scope and taking optimum decisions: starting from environment perception the agent constructs beliefs, then based on beliefs a list of options to act are generated: desires. Desires are filtered and become intentions that can be transformed into actions which represent the key point to obtain the process results.

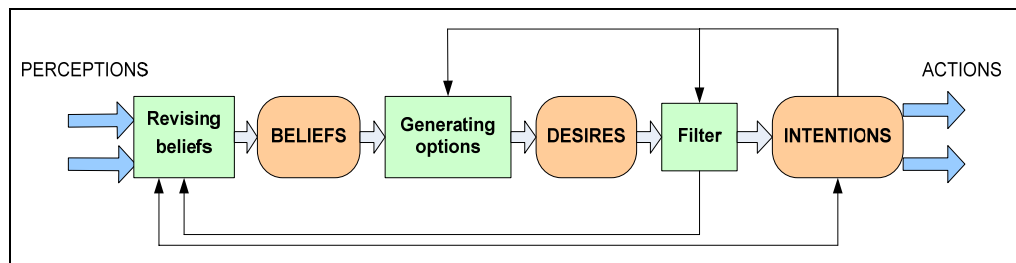


Figure 4. BDI classical diagram

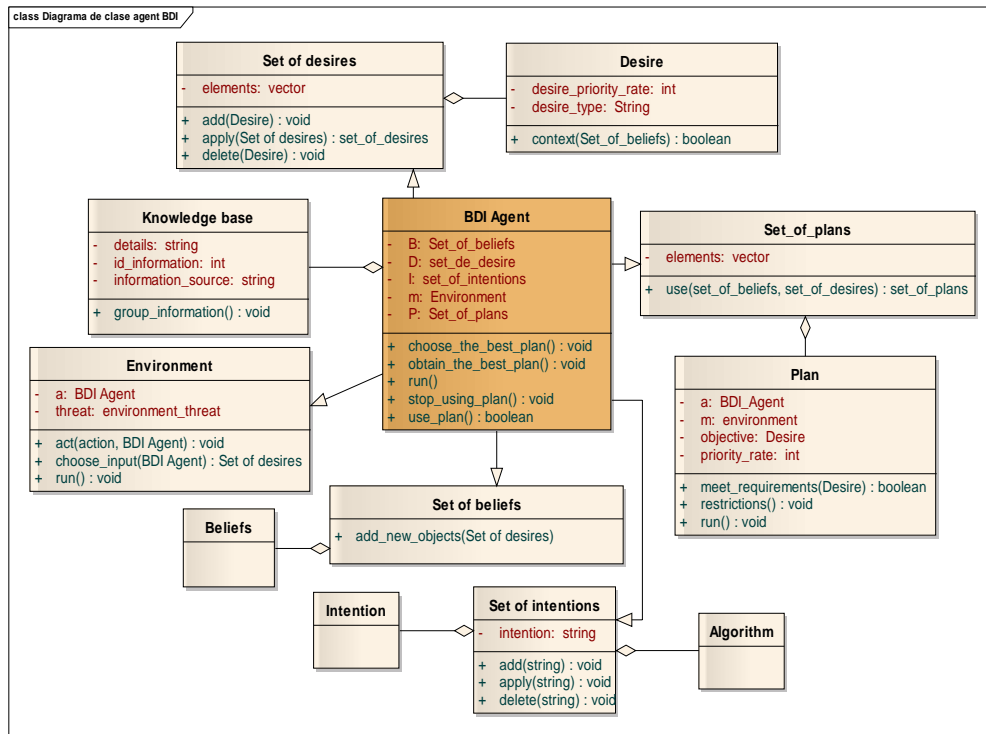
The class diagram for a BDI agent is shown in Figure 5. One criticism of the BDI model has been that it is not well-suited to certain types of behavior. In particular, the basic BDI model appears to be inappropriate for building systems that must learn and adapt their behavior – and such systems are becoming increasingly important. Moreover, the basic BDI model gives no architectural consideration to explicitly multi-agent aspects of behavior.

More recent architectures, such as InteRRaP and Touring Machine, do explicitly provide for such behaviors at the architectural level. InteRRaP (J.P. Müller and M. Pischel, 1993) derives from BDI with the following three improvements:

- a set of hierarchical control layers;
- a knowledge base that supports the representation of different abstraction levels of knowledge;
- a well-defined control structure that ensures coherent interaction among the control layers.

Touring Machine (Ferguson, 1992) architecture blends sophisticated and simplified control features on multi layers: Reactive Layer, Planning Layer, Control Layer and was designed by integrating a collection of both deliberative and non-deliberative agent control capabilities. Both architectures, InteRRaP and Touring

Machine cope with dynamic environments which is a great advantage against BDI architecture.



Source: Genco, 2007.

Figure 5. A general BDI Agent diagram

BDI architecture has several disadvantages that require improvements regarding:

- Learning process: BDI agents lack any specific mechanisms within the architecture to learn from past behaviour and adapt to new situations.
- Only three attitudes for agent behaviour: Classical decision theorists and planning researches question the necessity of having all three attitudes, distributed AI researches question whether the three attitudes are sufficient.
- Incomplete logic coverage: The multi-modal logics that underlie BDI (that do not have complete theoretical basis and are not efficiently computable) have little relevance in practice.
- Multiple Agents specifications: In addition to not explicitly supporting learning, the framework may not be appropriate to learning behavior. Further, the BDI model does not explicitly describe mechanisms for interaction with other agents and integration into a multi-agent system.

- Explicit Goals: Most BDI implementations do not have an explicit representation of goals.
- Anticipation: The architecture does not have (by design) any look-ahead deliberation or forward planning. This may not be desirable because adopted plans may use up limited resources, actions may not be reversible, task execution may take longer than forward planning, and actions may have undesirable side effects if unsuccessful.

4. An improved BDI architecture for project scheduling

We consider the Knowledge Base as very important for the agent behavior. Its behavior is influenced by perception and influences beliefs. *Knowledge Base*, as long term memory for all agents becomes a necessity to be added as long as project planning and scheduling process as repetitive activity for a project manager and therefore dependent to human experience and knowledge. For perceptions, we use a short term memory called Perception Buffer, which can be accessed by all agents in order to share their information about the environment. The *revising beliefs* process becomes more complex by extracting information through algorithms from perception and matching it with knowledge patterns stocked in Knowledge Base. Filter functionality is represented by constraints regarding agent goal and influences its behavior as limiting and directing space solution search. Constraints can be seen both on agent and system level (Figure 6). Regarding Filter, we consider two types of agents: time constrained and cost constrained, this influences also Knowledge Base as it should reflect specific type of agent. Also we have to insert an important process: *Anticipation* between *Intentions* and *Actions*. Anticipation, as algorithm, uses knowledge patterns to predict environment and possible course of actions and therefore is a rational characteristic for scheduling agent. Another link we have to make is between *Perceptions* and *Anticipation* because action-reaction mechanism is the real method to adjust agent behavior and maintain feedback control.

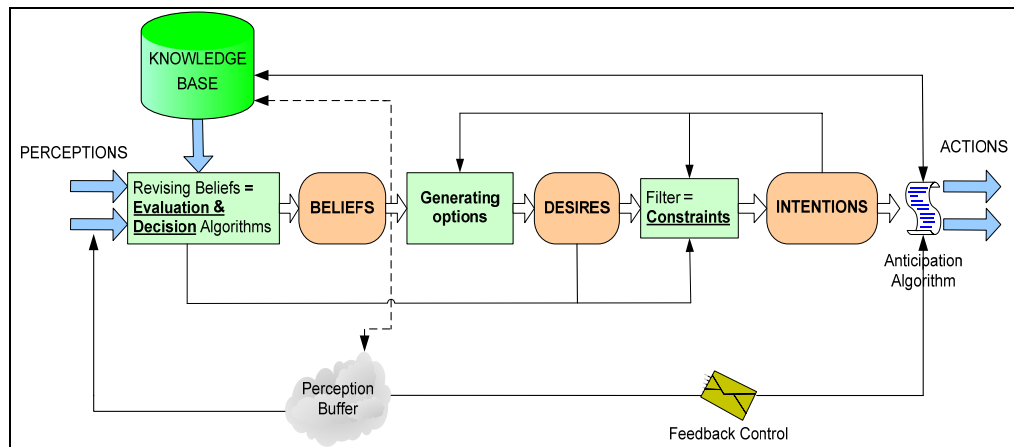


Figure 6. The BDI improved architecture

An important characteristic for a scheduling agent is the reasoning type. We consider practical reasoning to be used more than theoretical reasoning. Practical reasoning is directed towards actions and it is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent's beliefs. (Bratman, 1987). On the other side, theoretical reasoning is directed towards beliefs and has an indirect impact on solution quality; therefore we focus on the previous type of reasoning as it offers more obvious results.

Intelligent Resource-Bounded Machine Architecture (IRMA) developed on Bratman's ideas and BDI structures is an example of practical reasoning implementation. Comparing with IRMA architecture, our Anticipation Algorithm corresponds to IRMA's Opportunity Analyzer, but it has more features and links to other modules of the diagram (as Knowledge Base and Perception).

We correlate Anticipation with Attention (Figure 7). Posner and Peterson (1990) propose that attention comprises a system of anatomical regions which can be divided into the networks of alerting, orienting and executive control. In Neuroscience, Alerting performs the function of achieving and maintaining a vigilant state; Orienting refers to selective visual-spatial attention; and Executive control involves monitoring and resolving conflict in the presence of conflicting information. Anticipation Algorithm has inputs from Knowledge Base and Perception Buffer and generates a course of more efficient Actions with help of the three mentioned mechanism: alerting, orienting and executive control.

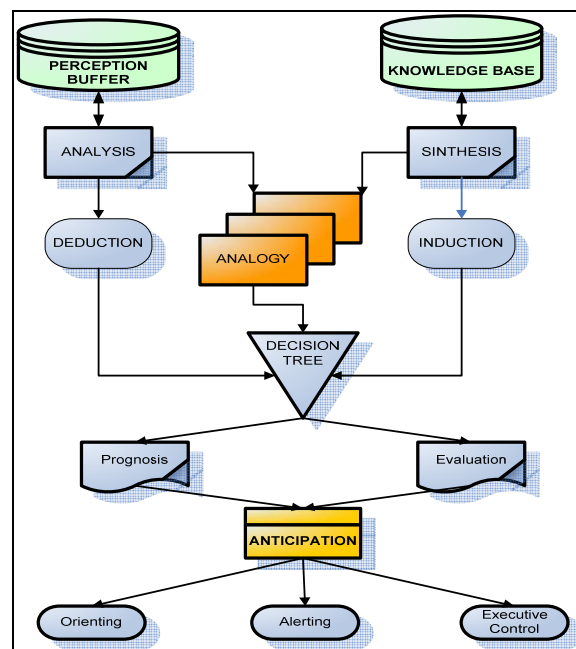


Figure 7. The anticipation mechanisms

The Figure 8 presents the three main internal components of a rational agent we want to create to solve the scheduling problem. The difference between time constrained and cost constrained agents appears at memory structure: time based or cost based structures and types of data passed through interfaces and internally evaluated. A scheduling agent should be able to record a history of all changes made to its data structures, including the identity of the agents he contacts. Queries could also be included in the history. Moreover, as the system is resource bounded, it is desirable to cache important information in Agent Self Memory rather than re-compute it from base perceptual data. Emotions can be considered a filter for agent's reactions and has also connections with memory; strong emotions as adversity to risk, preference to specific actions are stored in Agent Self Memory. Message Dispatcher allows parallel execution of tasks between agent's components and distributes alerting messages to appropriate solver.

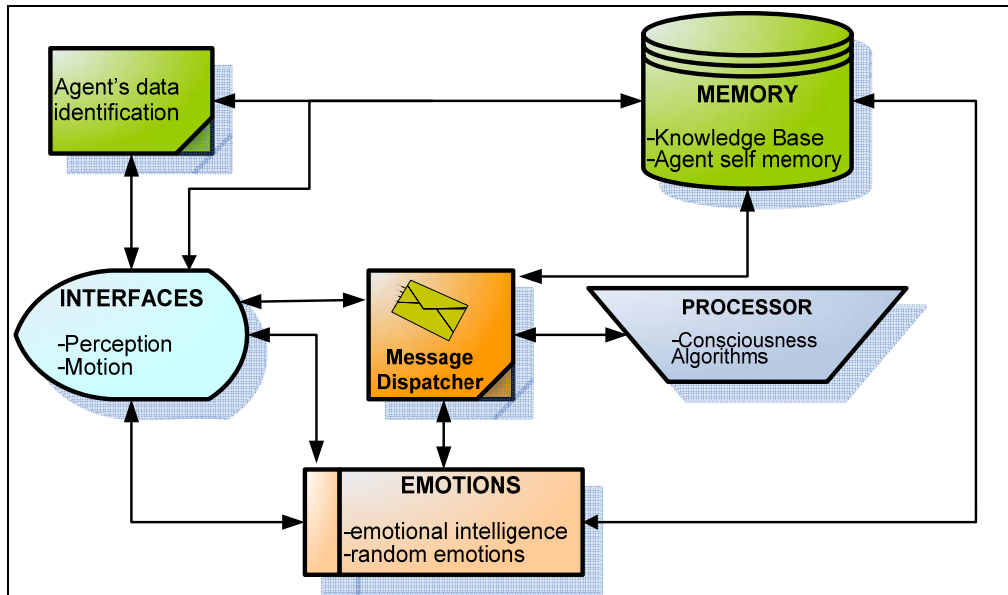


Figure 8. The Scheduling Agent Components

In order to design MAS for solving the proposed scheduling problem, we consider specialized agents which communicate directly by using cooperation on the same level of constraints: time or cost and in case of conflicts using negotiation mediators as showed in the Figure 9.

We consider three types of mediators at Time Constrained Level, Cost-Constrained Level and at System Level. The System Level Mediator has also the role to evaluate system performance to achieve the optimization goal. In fact, the mentioned levels for mediators reflect possible nodes of conflicts between agents

where negotiators are needed. The strategic-negotiation model we intend to implement is based on Rubinstein's model of alternating offers (Rubinstein, 1982). Agent negotiations without efficient algorithms can be very costly, lengthy and error-generator, this can be resolved through concessions and making new alternative offers between agents (Pruitt, 1993). The objective of the proposed negotiation model is to obtain a win-win solution. This is achieved by iterative bilateral interactions on concession and searching for new alternatives. The process of negotiation may be of many different forms, such as auctions, protocols in the style of the contract net, and argumentation, but it is unclear just how sophisticated the agents or the protocols for interaction must be for successful negotiation in different contexts. Jennings (1993) identified three broad topics for research on negotiation. First, the *negotiation protocols* are the set of rules that govern the interaction. This covers, the permissible types of participants (e.g., the negotiators and relevant third parties), the negotiation states (e.g., accepting bids, negotiation closed), the events that cause state transitions (e.g., no more bidders, bid accepted), and the valid actions of the participants in particular states (e.g., which can be sent by whom, to whom and at when). Second, *negotiation objects* are the range of issues over which agreement must be reached. These may single issues, such as price, or multiple issues relating to price, quality, timing, etc. Finally, the *agents' reasoning models* provide the decision making apparatus by which participants attempt to achieve their objectives. The sophistication of the model is determined by the protocol used, the nature of the negotiation object, and the range of operations that can be performed on it.

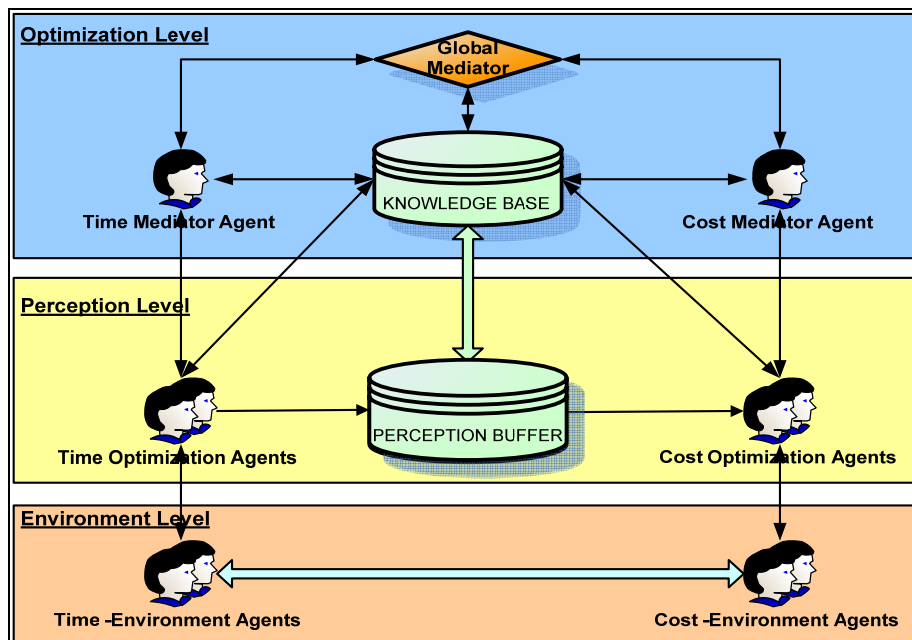


Figure 9. The MAS negotiation diagram

Applications such as Auction Bot (2002) and Kasbah (Chavez and Maes, 1996) use agents to perform negotiation tasks. However, agents in such applications negotiate solely on price and a deal is made based on the cheapest product offered by the vendor, without evaluating other issues such as guarantee, time of delivery or shipping fees. For example, this problem was experienced by the Bargain Finder (2002) system. The system operates as an intermediary between the buyer and seller to search the internet for information on products from vendor.

There are two patterns of interactions in auctions. The most common are one-to-many auction protocols (Gimenez-Funes, Godo and Rodriguez-Aguilar, 1998; Sandholm, 1993) where one agent initiates an auction and a number of other agents can bid in the auction, or many-to-many auction protocols (Wurman, Walsh, and Wellman, 1998) where several agents initiate an auction and several other agents can bid in the auction. Given the pattern of interaction, the first issue to determine is the type of protocols to use in the auction (Klemperer, P., 1999). Given the protocol, the agents need to decide on their bidding strategy. There are several types of one-to-many auctions which are used, including the English auction, first-price sealed-bid auction, second-price sealed-bid (Vickery auction), and the Dutch auction.

In systems composed of multiple autonomous agents, negotiation is a key form of interaction that enables groups of agents to arrive at a mutual agreement regarding some belief, goal or plan, for example. Particularly because the agents are autonomous and cannot be assumed to be benevolent, agents must *influence* others to convince them to act in certain ways, and negotiation is thus critical for managing such inter-agent dependencies. The process of negotiation may be of many different forms, such as auctions, protocols in the style of the contract net, and argumentation, but it is unclear just how sophisticated the agents or the protocols for interaction must be for successful negotiation in different contexts.

Conry, Kuwabara, Lesser and Meyer (1991) suggest multi-stage negotiation to solve distributed constraint satisfaction problems when no central planner exists. Moehlman, Lesser and Buteau (1992) use negotiation as a tool for distributed planning: each agent has certain important constraints, and it tries to find a feasible solution using a negotiation process. Lander and Lesser (1992) use a negotiation search, which is a multi-stage negotiation as a means of cooperation while searching and solving conflicts among the agents.

Rosenschein and Zlotkin (1994) identified three distinct domains where negotiation is applicable for MAS and found a different strategy for each domain:

- Task-Oriented Domain: Finding ways in which agents can negotiate to come to an agreement, and allocating their tasks in a way that is beneficial to everyone;
- State-Oriented Domain: Finding actions which change the state of the “world” and serve the agents’ goals;
- Worth-Oriented Domain: Same as (2) above, but, in this domain, the decision is taken according to the maximum utility the agents gain from the states.

Sycara (1990) presented a model of negotiation that combines case-based reasoning and optimization of multi-attribute utilities. In her work agents try to influence the goals and intentions of their opponents. Kraus and Lehmann (1995) developed an automated Diplomacy player that negotiates and plays well in actual games against human players. Sierra, Faratin and Jennings (1997) present a model of negotiation for autonomous agents to reach agreements about the provision of service by one agent to another. Their model defines a range of strategies and tactics, distilled from intuition about good behavioral practice in human negotiation that agents can employ to generate offers and evaluate proposals. Zeng and Sycara (1998) consider negotiation in a marketing environment with a learning process in which the buyer and the seller update their beliefs about the opponent's reservation price using the Bayesian rule.

5. The first implementation results

The proposed architecture is implemented using the Jade platform, version 3.7. Jade is a framework developed in Java that offers facilities for the multi agent systems implementation. Platform development began in 2001 and now allows coordination of agents in a system using FIPA standard. The platform has a Main Container (main location of agents) that contains two dedicated agents called AMS (Agent Management System) and DF (Directory Facilitator). The platform also offers various facilities including Sniffer agent used to describe communication between agents. The sequence diagram of a scenario for the negotiation process attended by three cost agent and three time agent is illustrated in Figure 11. This is achieved by using the facility "Sniffer" in the JADE plat for minter face (Figure 10).

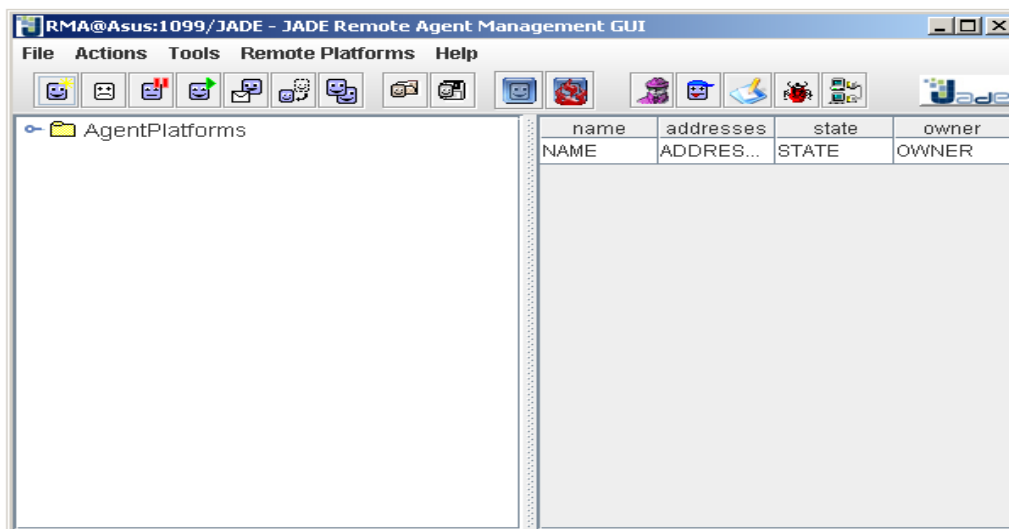


Figure 10. The JADE platform interface

A multi-agent system with application in project scheduling

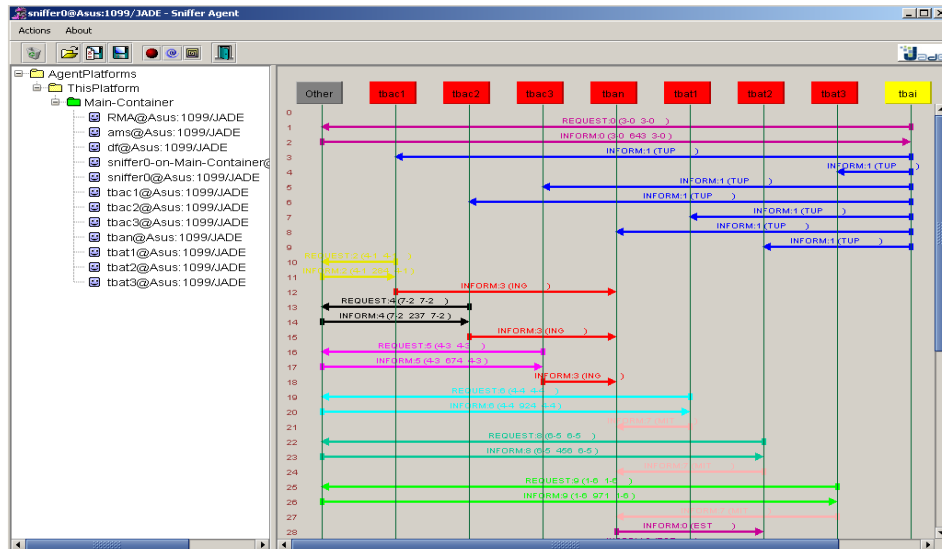


Figure 11. The JADE Sniffer agent

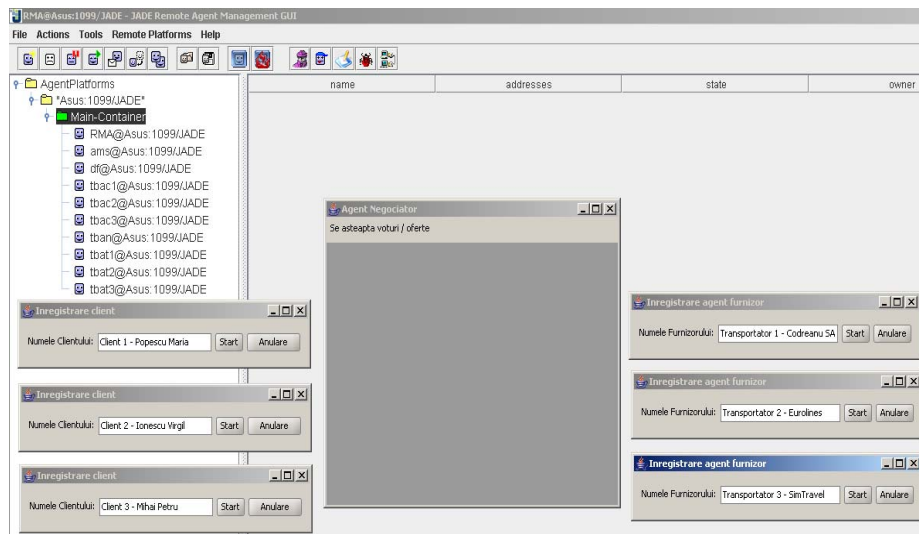


Figure 12. The main container agents

In Figure 12 is shown the agents used in order to optimize the cost and time resources, as well as the agent mediator/negotiator who will be take into account in the process of mediation/negotiation of resources used (time and cost, respectively). On the left side are shown all the agents that are contained in the main container: RMA (the Remote Monitoring Agent, which's an agent in JADE platform that provides user swith the graphical interface of its facilities), AMS (the Agent Management System),

DF (the Directory facilitator), tbac (the optimizing cost agent), that (the optimizing time agent), than (the mediator/negotiator agent). The negotiator/mediator agent waits for there quests from the cost and time agents. The negotiator agent could have an important role in the decision process of a project manager. The agent uses specific algorithms that offer the best results regarding the use and combination of resources used.

6. Conclusions

This paper approaches the BDI architecture implementation for time constrained scheduling problem extended with cost constraints. We observed the need for improvements at the architecture level regarding Knowledge Base containers and mechanisms to act more efficiently (as Anticipation). The improvements can be observed also at reactive architectures as Brooks's architecture. Besides reactivity to environment events, the scheduling agents have pro-active (anticipation) characteristic and social skills (interact with other agents with the same or different structure). The emotional component is seen from two perspectives: deterministic emotional intelligence and irrational actions, randomly generated in established limits.

Scheduling problem particularities require specialization among agents in the multi-agent system in time constrained, cost constrained and mediator agents. The main improvement to Brooks's architecture represents the anticipation mechanism which increases individual performance to solve the scheduling problem and uses information about environment from all agents: Knowledge Base and Perception Base.

The agent structure is influenced by its behavior as long as internal data structure has to carry appropriate information according to its scope. Agent types derive from their roles and level of access to knowledge and limited actions, respectively: the environment agents (know environment constraints), the optimization agents (collaborate in order to solve the scheduling problem) and the specialized mediator agents (know global constraints for time and/or cost and solve conflicts between other agents). The further steps in our research would be to fully implement the proposed architecture on Jade platform.

References

- Bodea, Badea and Purnus (2010), "Complex Project Scheduling using Multi-Agent Methods: a Case Study for Research Projects", *Management & Marketing*, No. 3
- Bodea, C. and Niculescu, C.S. (2006), "Resource Leveling Using Agent Technologies", *The Proceedings of 20th IPMA World Congress on Project Management*, China Machine Press, pp. 599-605
- Bratman, M.E. (1999), *Intention, Plans, and Practical Reason*, CSLI Publications
- Brenner W., Zarnekow, R., Witting H. (1998). *Intelligent Software Agent*, Springer Verlag
- Brooks, R.A. (1986), „A robust layered control system for a mobile robot”, *IEEE Journal of Robotics and Automation*, Vol. 2, No. 1, pp. 14-23

- Chavez and P. Maes (1996), "Kasbah: an agent marketplace for buying and selling goods", in *Proceedings of the first international Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, April 1996
- Chapman, D., Agre, P. (1986), "Abstract reasoning as emergent from concrete activity", in Georgeff, M.P. and Lansky, A.L. (eds.), *Reasoning About Actions & Plans – Proceedings of the 1986 Workshop*, pp. 411-424, Morgan Kaufmann Publishers: San Mateo, CA
- Chen, S.-H. (2000), Toward an Agent-Based Computational Modeling of Bargaining Strategies in Double Auction Markets with Genetic Programming, in: *Proceedings of the Second International Conference on Intelligent Data Engineering and Automated Learning, Data Mining, Financial Engineering, and Intelligent Agents*, Springer-Verlag London, UK, ISBN 3-540-41450-9
- Cohen, P.R., Levesque, H.J. (1990), "Intention is choice with commitment", *Artificial Intelligence*, Vol. 42, No. 2-3, pp. 213-261
- Conry, S.E., Kuwabara, K., Lesser V.R., Meyer, R.A. (1991), "Multistage negotiation for distributed satisfaction", *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence*, Vol. 21, No. 6, pp. 1462-1477
- Davidsson, P. (2000), *Multi Agent Based Simulation: Beyond social simulation*, in Multi Agent Based Simulation (LNCS Vol. 1979), Springer Verlag
- Dawid, H. (1999), "On the Convergence of Genetic Learning in a Double Auction Market", *Journal of Economic Dynamics and Control*, No. 23
- Ferguson, A. (1992), *Touring Machines: Autonomous Agents with Attitudes*, Vol. 25, No. 5, May 1992, IEEE Computer Society Press Los Alamitos, CA, USA
- Durfee, E.H. (1999), "Practically coordinating", *AI Magazine*, Vol. 20, No. 1, pp. 99-116
- Durfee, E.H., Lesser, V. (1989), "Negotiating Task Decomposition and Allocation Using Partial Global Planning", in L. Gasser, M. Huhns (eds.), *Distributed Artificial Intelligence*, Vol. 2, pp. 229-244. San Francisco, Calif., Morgan Kaufmann
- Fikes, R. Nilsson, N. (1971), "STRIPS: A new approach to the application of theorem proving to problem solving", in *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI-71)*, pp. 608-620
- Genco A. (2007), *Mobile Agents*, WitPress, UK
- Gimenez-Funes, E., Godo, L., Rodriguez-Aguilar J.A. (1998), „Designing bidding strategies for trading agents in electronic commerce”, in: *ICMAS98*, pp. 136-143, Paris
- Heckbert, S. (2009), Experimental economics and agent-based models, *18th World IMACS/MODSIM Congress, Cairns, Australia 13-17 July 2009*, available at: <http://mssanz.org.au/modsim09>
- Hurink J.L., Kok A.L., Paulus J.J., Schutten J.M.J. (2009), *Time-constrained project scheduling with adjacent resources*, Enschede, Netherlands
- Hussain, F., Wood, S. (2009), Modeling the Performance of Children on the Attentional Network Test, in A. Howes, D. Peebles, R. Cooper (eds.), *9th International Conference on Cognitive Modeling – ICCM09*, July 2009, Manchester, UK: University of Manchester, pp. 211-216
- Jennings, N.R. (1993), "Commitments and conventions: the foundation of coordination in multi-agent systems", *The Knowledge Engineering Review*, Vol. 8, No. 3, pp. 223-250
- Jennings, N.R. et. Al. (2001), Automated negotiation: prospects, methods and challenges, *International Journal of Group Decision and Negotiation*, Vol. 10, No. 2, pp. 199-215
- Kaelbling, L.P., Rosenschein, S.J. (1990), "Action and planning in embedded agents", in Maes, P. (ed.), *Designing Autonomous Agents*, pp. 35-48, The MIT Press: Cambridge, MA
- Klemperer, P. (1999), "Auction theory: A guide to literature", *Journal of Economic Surveys*, Vol. 13, No. 3

- Kolisch, R., Padman, R. (2001), "An integrated survey of deterministic project scheduling", *Omega*, 29, pp. 249-272
- Kraus, S. (2001), *Strategic Negotiation in Multi-Agent Environments*, Cambridge, MA: MIT Press
- Kraus S., Lehmann D. (1995), "Designing and building a negotiating automated agent", *Computational Intelligence*, Vol. 11, No. 1, pp. 132-171
- Lander S.E., Lesser, V.R. (1992), Customizing distributed search among agents with heterogeneous knowledge, in *Proc. first int. conf. on Information Knowledge Management*, pp. 335-344, Baltimore
- Lee, J., Kim, Y. (1996), "Search heuristics for resource constrained project scheduling", *Journal of the Operational Research Society*, Vol. 47, No. 5, pp. 678-689
- Moehlman, T., Lesser, V. and Buteau, B. (1992), "Decentralized negotiation: An approach to the distributed planning problem", *Group Decision and Negotiation*, Vol. 2, pp. 161-191
- Müller, J.P., Pischel, M. (1993), *The agent architecture interRRaP: Concept and application*, German Research Center for Artificial Intelligence
- Palensky P. (1999), *The Convergence of Intelligent Software Agents and Field Area Networks*, IEEE 0-7803-5670-5/99
- Posner, M.I., Peterson S.E. (1990), "The attention system of the human brain", *Annual Review of Neuroscience*, Vol. 13, pp. 25-42
- Pruitt, D. (1993), *Negotiation in Social Conflict*, Open University Press, Buckingham
- Pynadath, D., Tambe, M. (2002), "Multi-agent teamwork: analyzing key teamwork theories and models", in Castelfranchi, C., Johnson, L. (eds.), *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Vol. 2, pp. 873-880
- Rao, A., Georgeff, M. (1995), BDI-Agents: From Theory to Practice, in *Proceedings of the First International Conference on Multiagent Systems (ICMAS '95)*, San Francisco
- Rosenschein, J., Zlotkin, G. (1994), *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*, Cambridge MA: MIT Press
- Rubinstein, A. (1982), "Perfect equilibrium in a bargaining model", *Econometrica*, Vol. 50, No. 1, pp. 97-109
- Russell, S.J., Norvig, P. (2003), *Artificial Intelligence: a Modern Approach*, 2nd edition, Prentice Hall,
- Sandholm T. (1993), "An implementation of the contract net protocol based on marginal cost calculations", in *Proc. of AAAI-93*, pp. 256-262, Washington D.C.
- Seah, M., Sierhuis, W.J. Clancey (2011), *Multi-agent Modelling and Simulation Approach for Design and Analysis of MER Mission Operations*, NASA papers
- Sierra, C., Faratin, P., Jennings, N. (1997), "A service-oriented negotiation model between autonomous agents", in: *Proc. 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-97)*, pp. 17-35, Ronneby, Sweden
- Sycara, K.P. (1990), "Persuasive argumentation in negotiation", *Theory and Decision*, Vol. 28, pp. 203-242
- Wooldridge, M., Jennings, N.R. (1994), "Formalizing the cooperative problem solving process", in *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence (IWDAI-94)*, pp. 403-417, Lake Quinalt, WA
- Wood, S. (1990), *Planning in a Rapidly Changing Environment*, PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK
- Wurman, P.R., Walsh, W.E., Wellman, M.P. (1998), "Flexible double auctions for electronic commerce: Theory and implementation", *Decision Support Systems*, Vol. 24, pp. 17-27
- Zeng, D., Sycara, K. (1998), Bayesian learning in negotiation, *International Journal of Human-Computer Studies*, Vol. 48, pp. 125-141